

AMENDMENTS TO THE CLAIMS

1. (CURRENTLY AMENDED) A computer-implemented method that authors information stored in a document, wherein a client computer processes the information at run-time, comprising the steps of:

receiving information about a first control that has both property types and property values that define how the first control appears in the document;

wherein a software application contains the first control and whose application's extensibility interfaces do not natively support a design-time control;

~~wherein the first control generates both design-time information and run-time information, wherein the run-time information is processed at run-time, wherein the design-time information is not used during run-time;~~

creating a second control as a design-time control whose property types substantially match the property types of the first control;

wherein the second control is created by:

(i) receiving at least a portion of the run-time information of the first control;

(ii) parsing the run-time information in order to determine the property values of the first control;

(iii) determining type of design-time control to create as the second control based upon the determined property values;

generating an interface for modifying by the user of the property values of the second control; and

using the second control's modified property values to update the property values of the first control in the document by converting the second control's modified property values into run-time information to be used within the document.

2. (ORIGINAL) The method of claim 1 wherein the design-time information is used during authoring of the information.

3. (CANCELLED)

4. (CANCELLED)

5. (CURRENTLY AMENDED) The method of claim 41 further comprising the steps of:

providing a data communication connection to the software application through at least one of the extensibility interfaces;

receiving information about the first control through the data communication connection; and

sending the second control's modified property values to the software application through the data communication connection in order to update the property values of the first control in the document.

6. (ORIGINAL) The method of claim 1 wherein a graphic image represents the first control in the document.

7. (ORIGINAL) The method of claim 6 further comprising the step of:

receiving information about the first control after the graphic image of the first control has been activated by a user in order to edit the property values of the first control.

8. (CURRENTLY AMENDED) The method of claim 1 further comprising the step of:

receiving an identifier to indicate type of ~~Design Time Control~~ design time control to create as the second control, wherein the identifier is received when the property values of the first control are empty.

9. (ORIGINAL) The method of claim 8 wherein the identifier is retrieved from an operating system registry.

10. (CURRENTLY AMENDED) The method of claim 1 further comprising the steps of:

receiving the property values of the first control, wherein at least one of the property values is not empty; and

determining type of design time control ~~Design Time Control~~ to create as the second control based upon the property values.

11. (CANCELLED)

12. (CURRENTLY AMENDED) The method of claim 11 wherein the first control is a tag that is inserted into the document; wherein the run-time information includes Hypertext Markup Language (HTML) run-time instructions.

13. (CURRENTLY AMENDED) The method of claim 1 wherein the run-time information includes Hypertext Markup Language (HTML) run-time instructions, said method further comprising the steps of:

receiving the HTML run-time instructions of the first control;
parsing the HTML run-time instructions to obtain a properties map, metadata tag text, a control class identifier, a program identifier, and inner HTML information; and
determining type of design time control ~~Design-Time-Control~~ to create as the second control based upon the parsed HTML run-time instructions.

14. (ORIGINAL) The method of claim 1 further comprising the step of:

creating a hidden window to store the second control.

15. (ORIGINAL) The method of claim 1 further comprising the step of:

generating a computer-human interface for modifying the property values of the second control.

16. (CURRENTLY AMENDED) The method of claim 15 wherein the run-time information includes Hypertext Markup Language (HTML) run-time instructions, said method further comprising the steps of:

receiving the HTML run-time instructions of the first control;

parsing the HTML run-time instructions to obtain a properties map, metadata tag text, a control class identifier, a program identifier, and inner HTML information;

determining type of design time control ~~Design-Time-Control~~ to create as the second control based upon the parsed HTML run-time instructions; and

determining what properties to display in the computer-human interface based upon the parsed HTML run-time instructions.

17. (CANCELLED)

18. (ORIGINAL) The method of claim 1 further comprising the step of:

converting the second control's modified property values into Hypertext Markup Language (HTML) run-time instructions to be used within the document.

19. (ORIGINAL) The method of claim 1 wherein the client computer receives the document over a network in order to processes the information.

20. (ORIGINAL) The method of claim 19 wherein the network is selected from the group consisting of local area networks, wide area networks, global networks, and combinations thereof.

21. (ORIGINAL) The method of claim 1 wherein the document is an Hypertext Markup Language (HTML) document.

22. (CURRENTLY AMENDED) A computer-implemented apparatus that authors information stored in a document, wherein a client computer processes the information at run-time, comprising:

a ~~Design~~design-Time-time Control-control wrapper module that receives information about a first control, wherein the first control has both property types and property values that define how the first control appears in the document;

wherein a software application contains the first control and whose application's extensibility interfaces do not natively support a design-time control;

wherein the first control generates both design-time information and run-time information, wherein the run-time information is processed at run-time, wherein the design-time information is not used during run-time;

a window connected to the ~~Design~~design-Time-time Control-control wrapper within which a second control is created, wherein the second control's property types substantially match the property types of the first control; and

wherein the second control is created by:

(i) receiving at least a portion of the run-time information of the first control;

(ii) parsing the run-time information in order to determine the property values of the first control;

(iii) determining type of design-time control to create as the second control based upon the determined property values;

a properties interface connected to the window to modify the property values of the second control,

whereby the second control's modified property values are used to update the property values of the first control in the document by converting the second control's modified property values into run-time information to be used within the document.

23. (ORIGINAL) The apparatus of claim 22 wherein the design-time information is used during authoring of the information.

24. (CANCELLED)

25. (CANCELLED)

26. (CURRENTLY AMENDED) The apparatus of claim ~~25~~22 further comprising:

a run-time translator module connected to the ~~Design-Time Control~~design time control wrapper module, wherein the run-time translator includes a data communication connection to the software application through at least one of the extensibility interfaces,

wherein the run-time translator module receives information about the first control in order to provide the information to the design time control ~~Design-Time Control~~ wrapper module.

27. (ORIGINAL) The apparatus of claim 22 wherein a graphic image represents the first control in the document.

28. (CURRENTLY AMENDED) The apparatus of claim 22 wherein the design time control ~~Design-Time Control~~ wrapper module receives an identifier to indicate type of design time control ~~Design-Time Control~~ to create as the second control, wherein the identifier is received when the property values of the first control are empty.

29. (ORIGINAL) The apparatus of claim 28 wherein the identifier is from an operating system registry.

30. (CURRENTLY AMENDED) The apparatus of claim 22 wherein the design time control ~~Design-Time-Control~~-wrapper module receives property values of the first control, wherein at least one of the property values is not empty, wherein the design time control ~~Design-Time-Control~~-wrapper module determines type of design time control ~~Design-Time-Control~~ to create as the second control based upon the received property values.

31. (CURRENTLY AMENDED) The apparatus of claim 22 wherein the design time control ~~Design-Time-Control~~-wrapper module parses the run-time information of the first control in order to determine the property values of the first control, said apparatus further comprising:

a properties data structure to store the property values,

wherein type of design time control ~~Design-Time-Control~~ to create as the second control is based upon the stored property values.

32. (ORIGINAL) The apparatus of claim 31 wherein the run-time information includes Hypertext Markup Language (HTML) run-time instructions.

33. (CURRENTLY AMENDED) The apparatus of claim 32 further comprising:

a properties map data structure connected to the design time control
~~Design-Time-Control~~-wrapper module to store names and values of the properties parsed
from the HTML run-time instructions,

wherein the interface includes the property names and values stored in the
properties map data structure.

34. (CURRENTLY AMENDED) The apparatus of claim 32 further comprising:

a metadata tag text data structure connected to the design time control
~~Design-Time-Control~~-wrapper module to store a header portion contained in the HTML
run-time instructions.

35. (CURRENTLY AMENDED) The apparatus of claim 32 further comprising:

a control class identification data structure connected to the design time
control ~~Design-Time-Control~~-wrapper module to store a control classification identifier
contained in the HTML run-time instructions.

36. (CURRENTLY AMENDED) The apparatus of claim 32 further comprising:

a program identifier data structure connected to the design time control
~~Design-Time-Control~~-wrapper module to store a program identifier contained in the
HTML run-time instructions,

wherein the stored program identifier is used to determine what type of
design time control ~~Design-Time-Control~~ to create within the window.

37. (CURRENTLY AMENDED) The apparatus of claim 32 further comprising:

an inner HTML data structure connected to the design time control
~~Design-Time-Control~~-wrapper module to store substantially a complete HTML text
representation of the first control.

38. (ORIGINAL) The apparatus of claim 22 wherein the window is a hidden window to
store the second control.

39. (ORIGINAL) The apparatus of claim 22 wherein the properties interface is a
computer-human interface that modifies the property values of the second control.

40. (CURRENTLY AMENDED) The apparatus of claim 39 wherein the run-time
information includes Hypertext Markup Language (HTML) run-time instructions, said
apparatus further comprising:

means for receiving the HTML run-time instructions of the first control;
means for parsing the HTML run-time instructions to obtain a properties
map, metadata tag text, a control class identifier, a program identifier, and inner HTML
information;

means for determining type of design time control ~~Design-Time-Control~~ to
create as the second control based upon the parsed HTML run-time instructions; and
means for determining what properties to display in the computer-human
interface based upon the parsed HTML run-time instructions.

41. (ORIGINAL) The apparatus of claim 22 further comprising:

means for converting the second control's modified property values into run-time information to be used within the document.

42. (ORIGINAL) The apparatus of claim 22 further comprising:

means for converting the second control's modified property values into Hypertext Markup Language (HTML) run-time instructions to be used within the document.

43. (ORIGINAL) The apparatus of claim 22 wherein the client computer receives the document over a network in order to processes the information.

44. (ORIGINAL) The apparatus of claim 43 wherein the network is selected from the group consisting of local area networks, wide area networks, global networks, and combinations thereof.

45. (ORIGINAL) The apparatus of claim 22 wherein the document is an Hypertext Markup Language (HTML) document.

46. (CURRENTLY AMENDED) The apparatus of claim 22 wherein the design time control ~~Design-Time-Control~~-wrapper module handles a plurality of different types of design time control ~~Design-Time-Controls~~ that are used within the document.

47. (NEW) A computer-implemented method that authors information stored in a document, wherein a client computer processes the information at run-time, comprising the steps of:

receiving information about a first control that has both property types and property values that define how the first control appears in the document,

wherein the first control includes run-time information, wherein the run-time information is processed at run-time;

creating a second control whose property types substantially match the property types of the first control;

wherein the second control generates both design-time information and run-time information, wherein the run-time information is processed at run-time, wherein the design-time information is not used during run-time;

generating an interface for modifying the property values of the second control; and

using the second control's modified property values to update the property values of the first control in the document.